REMARKS

Claims 1-24 were pending. Claims 1, 5, 9, 14, 20, and 24 are independent. Claims 1, 5, 9, 14, and 20 are amended. Claim 25 is added. No new matter is added. This response is being filed with a Request for Continued Examination. Reconsideration of the action mailed October 30, 2007, is respectfully requested in light of the foregoing amendments and the following remarks.

The Examiner rejected claims claim 24 under 35 U.S.C. § 102(e) as allegedly anticipated by U.S. Patent No. 6,330,588 ("Freeman"). The Examiner rejected claims 1-23 under 35 U.S.C. § 103(a) as allegedly unpatentable over Jansen et al. "NIST Special Publication 800-19—Mobile Agent Security" ("Jansen") in view of U.S. Patent No. 6,233,601 ("Walsh") and Freeman. Applicant respectfully traverses the rejections.

Section 102 Rejections

Claim 24 stands rejected over Freeman. Claim 24 is directed to a method that includes determining whether a first host is an untrusted host and, when the first host is an untrusted host, replacing code in the jumping application that implements a particular behavior with a piece of code that implements the particular behavior in the jumping application so that the jumping application has the particular behavior when it is executed by the second host for each jump of the jumping application between hosts.

The Examiner states that Freeman discloses replacing code of an untrusted hosts at col. 14, lines 7-18. Applicant respectfully disagrees. The cited portion of Freeman reads, in pertinent part, as follows:

> So implementing these mechanisms (and components) enables (a) the trusted resource to operate substantially protected from corruption and/or (b) the trusted resource's implicated mechanism(s) to be compared to, e.g., firmware copies from time to time so that, if any corruption is discovered, corrective measures can be invoked. Typical corrective measures include repairing software, e.g., by reloading objects, by deleting any extraneous code, by application of code comparison and correction algorithms or similar routines, and/or by other

> conventional approaches, whether automatically or by human initiation, such as by a system operator.

The cited portion of Freeman discloses using a resource to verify mobile software agents. The verification identifies and detects corruption in the mobile agent to protect the mobile agent. The mobile agent can be corrected when corruption is identified. The correction includes repairing software by reloading an object, deleting code, and correcting code.

Thus, Freeman discloses correcting the mobile agent when a corruption is identified. However, claim 24 requires that code be replaced from a jumping application when the first host is an untrusted host.

Additionally, the Examiner asserts that_identifying corrupted code is equivalent to the host being untrusted. Applicant respectfully disagrees. In claim 24, the term "untrusted" refers to the host, not the code. Thus, trust, as recited in claim 24, is a property of the host and not the code. A host can be identified as untrusted with or without corrupted code. Similarly, a trusted host can become infected, for example by a virus, that results in corrupted code. Consequently, whether or not a host is determined to be trusted is not dependent upon identifying corruption in the code. Moreover, claim 24 additionally requires that particular code is only identified for replacement when the host is an untrusted host. Thus, claim 24 precludes corruption of code from being equivalent to the designation of a host as trusted or untrusted. Therefore, Freeman does not disclose or suggest replacing code for an untrusted host as required by claim 24. Applicant respectfully submits that claim 24 is in condition for allowance.

Section 103 Rejections

Claim 1 was rejected over Jansen, Walsh, and Freeman. Claim 1 is directed to a jumping application security console that includes instructions to replace code from the jumping application for each jump of the jumping application between hosts.

The Examiner states that Jansen and Walsh do not disclose the claimed replacing of code. Instead, the Examiner relies on Freeman as disclosing the claimed replacing of code from the jumping application at col. 13, lines 35-50 and col. 14, lines 7-18. Applicant respectfully disagrees.

The portion of Freeman at col. 13, lines 35-50 reads as follows:

> A security mechanism provides for securing selected portions of the computing environment from corruption. The security mechanism can be variously implemented. As an example, the security mechanism can be implemented to process received agents, so as to protect one or more of the software agent, the origin resource, the destination resource and the trusted resource from any corrupting influences potentially present in the agent itself, regardless of the source (e.g., sourced incident to exercising the computing environment). Toward accomplishing that end, the security mechanism typically is enabled to remove, disable or otherwise render ineffective any or all of a received agent that may be corrupted or corrupting. Such a mechanism preferably includes monitoring firmware and/or other hardware which is used to identify agents and other potentially executable code that may be corrupted.

The cited portion discloses that a security mechanism can be implemented to protect from corruption. In particular, the cited portion discloses that the security mechanism can remove or disable part of a mobile agent that has become corrupted or is corrupting. The cited portion, however, does not disclose or suggest replacing code each time a jumping application jumps between hosts, as required by claim 1.

Additionally, col. 14, lines 7-18 of Freeman reads, in pertinent part, as follows:

> So implementing these mechanisms (and components) enables (a) the trusted resource to operate substantially protected from corruption and/or (b) the trusted resource's implicated mechanism(s) to be compared to, e.g., firmware copies from time to time so that, <u>if any corruption is discovered, corrective measures can be invoked</u>. Typical corrective measures include repairing software, e.g., by reloading objects, by deleting any extraneous code, by application of code comparison and correction algorithms or similar routines, and/or by other conventional approaches, whether automatically or by human initiation, such as by a system operator. [Emphasis added]

The cited portion of Freeman discloses using a resource to verify mobile software agents. The verification identifies and detects corruption in the mobile agent to protect the mobile agent. The mobile agent can be corrected when corruption is identified. The correction includes repairing software by reloading an object, deleting code, and correcting code.

Thus, Freeman discloses correcting the mobile agent <u>when a corruption is identified</u>. However, claim 1 requires that code be replaced from a jumping application <u>each time the jumping application jumps between hosts</u>. Thus, code is replaced at each jump regardless of any corruption in the code. Freeman does not disclose or suggest replacing code each time the jumping application jumps between hosts. Therefore, Applicant respectfully submits that claim 1, as well as claims 2-4, which depend from claim 1, are in condition for allowance.

Claim 5 was rejected over Jansen, Walsh, and Freeman. Claim 5 is directed to a jumping application security console that includes replacing code from the jumping application that implements a first behavior with a piece of code from the database into the jumping application that implements the first behavior during each jump between hosts. As set forth above with respect to claim 1, the cited references do not disclose or suggest replacing code of a jumping application during each jump between hosts. Applicant respectfully submits that claim 5, as well as claims 6-8, which depend from claim 5, are in condition for allowance.

Claim 9 was rejected over Jansen, Walsh, and Freeman. Claim 9 is directed to a method that includes receiving a request for a piece of code each time a jumping application jumps between hosts and replacing code in the jumping application. For at least the same reasons as set forth above with respect to claim 1, claim 9, as well as claims 10-13, which depend from claim 9, are in condition for allowance.

Claim 14 was rejected over Jansen, Walsh, and Freeman. Claim 14 is directed to a jumping application security system that includes instructions to replace code from a jumping application with a piece of code from a database into the jumping application each time the jumping application jumps between hosts. For at least the same reasons as set forth above with respect to claim 1, claim 14, as well as claims 15-19, which depend from claim 14, are in condition for allowance.

Claim 20 was rejected over Jansen, Walsh, and Freeman. Claim 20 is directed to a server for a jumping application security system that includes instructions that replace code from a jumping application with a piece of code from a database into the jumping application each time the jumping application jumps from a first host to a second hosts. For at least the same reasons as set forth above with respect to claim 1, claim 20, as well as claims 21-23, which depend from claim 20, are in condition for allowance.

New Claims

Claim 25 is added. Claim 25 is directed to a jumping application security system that includes a security module having instructions to replace code from a jumping application with a piece of code from a database into the jumping application each time the jumping application jumps between hosts. The cited references of Jensen, Walsh, and Freeman do not disclose or suggest replacing code from a jumping application with a piece of code from a database into the jumping application each time the jumping application jumps between hosts. Applicant respectfully submits that claim 25 is allowable.

Conclusion

Applicant requests that all pending claims be allowed.

By responding in the forgoing remarks only to particular positions taken by the Examiner, Applicant does not acquiesce with other positions that have not been explicitly addressed. In addition, Applicants' arguments for patentability of a claim should not be understood as implying that no other reasons for the patentability of that claim exist. Finally, Applicant's decision to amend or cancel any claim should not be understood as implying that Applicant agrees with any positions taken by the Examiner with respect to that claim or other claims.

Please apply $105 for the additional independent claim fee, $405 for the RCE filing fee and any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: January 30, 2008

/Brian J. Gustafson/
Brian J. Gustafson
Reg. No. 52,978

Customer No. 26181
Fish & Richardson P.C.
Telephone: (650) 839-5070
Facsimile: (650) 839-5071